

TP2: Diseño de un lenguaje de programación

Profesora: Gabriela Sanroman

Comisión: K2032

Año: 2023

Grupo 10:

Nicolás R. Kalaydjian

Eitan Wajsberg

Nataly Sofia Rohling Insua

Nehuen Balian Amaros

Rocío Ochoa Charlín

Guido Marchi

Julian Alejandro Mosquera

Lenguaje GRITHON

Introducción: Presentaremos las categorías léxicas y sintácticas de un lenguaje de programación creado por nosotros mismos. Formalizaremos cada LR correspondiente a cada categoría léxica con expresiones regulares. También formalizaremos cada LIC con BNF. Todo esto con el fin de mostrar la gramática léxica y sintáctica de nuestro lenguaje de programación

Categorías Léxicas:

Operadores
Identificador
ConstanteEntera
Cadena
CaracteresPuntuación
PalabrasReservadas

Operadores = \+ | * | =

Identificador = Mayúscula(Mayúscula + GuiónBajo)*

GuiónBajo = _

Mayúscula = A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z

Minúscula = a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

ConstanteEntera = 0 | NroNoCero

NroNoCero = Signo DígitoNoCero (DígitoNoCero | 0)*

Signo = ε|-

DígitoNoCero = 1|2|3|4|5|6|7|8|9

Cadena = "(Minúscula + \s)*"

CaracteresPuntuación = \ (| \)

PalabrasReservadas = ent | cad | mostrar | ing

Categorías sintácticas:

programa: *listaDeclaraciones*_{op} *listaSentencias*_{op}

listaDeclaraciones: *sentDeclaración*
sentDeclaración listaDeclaraciones

sentDeclaración: *tipoDato* *Identificador* *valorAsignado*_{op}

valorAsignado: = *valorAAsignar*

valorAAsignar: uno de *ConstanteEntera* *Cadena* *exprAritmética*

tipoDato: uno de **ent** **cad**

(Si *tipoDato* es ent entonces *valorAAsignar* debe ser *ConstanteEntera* o *exprAritmética*. En caso contrario, será una *Cadena*)

listaSentencias: *sentIO*
sentIO listaSentencias

sentIO: uno de *sentIngreso* *sentSalida*

sentIngreso: **ing** *Identificador*

sentSalida: **mostrar** *salida*

salida: uno de *Identificador* *ConstanteEntera* *Cadena* *exprAritmetica*

(El identificador debe ser previamente declarado)

exprAritmética: *término*
exprAritmética + término

término: *factor*
*término * factor*

factor: *ConstanteEntera*
Identificador
(exprAritmética)

(El identificador debe ser previamente declarado como tipo ent)

PROGRAMA DE EJEMPLO N° 1:

cad HOLA = "hola"
ent NUMUNO = 12
ent NUMDOS = 5
ent NUMTRES = NUMUNO + NUMDOS
ent NUMCUATRO

mostrar "ingresar nro entero"
ing NUMCUATRO

mostrar NUMTRES + NUMCUATRO

DERIVACION DE PRUEBA

PROGRAMA DE EJEMPLO N° 2:

cad HOLA = "hola mundo"
mostrar HOLA

DERIVACION PROGRAMA 2:

<i>programa</i>
<i>listaDeclaraciones listaSentencias</i>
<i>listaDeclaraciones sentIO</i>
<i>listaDeclaraciones sentSalida</i>
<i>listaDeclaraciones mostrar salida</i>
<i>listaDeclaraciones mostrar identificador</i>
<i>listaDeclaraciones mostrar HOLA</i>
<i>sentDeclaracion mostrar HOLA</i>
<i>tipoDato Identificador valorAsignado mostrar HOLA</i>
<i>tipoDato Identificador = ValorAAsignar mostrar HOLA</i>
<i>tipoDato Identificador = Cadena mostrar HOLA</i>
<i>tipoDato Identificador = "hola mundo" mostrar HOLA</i>
<i>tipoDato HOLA = "hola mundo" mostrar HOLA</i>
<i>cad HOLA = "hola mundo" mostrar HOLA</i>

Conclusiones: Logramos desarrollar la sintaxis de un lenguaje de programación capaz de realizar operaciones de suma y multiplicación con números enteros, así como solicitar datos al usuario, mostrar datos y cadenas de texto por pantalla. Empleamos expresiones regulares para definir las diferentes categorías léxicas requeridas, como palabras reservadas, operadores, entre otros. Además, establecimos categorías sintácticas utilizando la notación BNF.